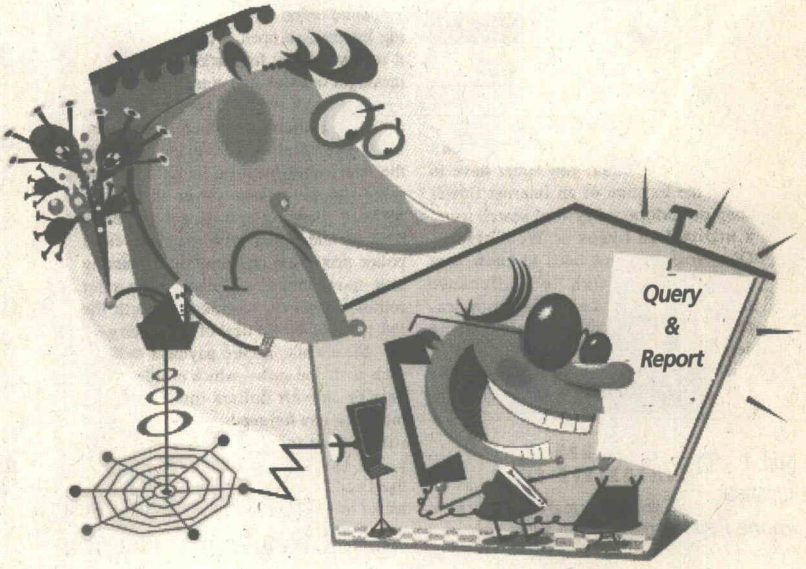
■ประพนธ์ อัศวภาณุวัฒน์



Query & Report ตอนที่ 1

คุยกันเรื่อง การนำ Query และ Report มาประยุกต์ใช้งานจริง

สวัสดีครับ การพัฒนาโปรแกรมในปัจจุบัน ต้องยอมรับว่าโปรแกรมส่วนใหญ่มีการทำงานในด้านการจัดเก็บข้อมูล และการออกรายงานเป็นส่วนประกอบที่สำคัญ โดยเฉพาะโปรแกรมทางด้านธุรกิจแล้ว แทบจะเป็นส่วนสำคัญที่นักพัฒนาระบบต้องคำนึงถึงเป็นอันดับแรกเลยทีเดียว

ในฉบับนี้ ตามสัญญาที่เคยให้ไว้ เราจะคุยกันเรื่อง การนำ Query และ Report มาประยุกต์ใช้งานจริง สำหรับผู้สนใจในส่วนของ Database Query เป็นพิเศษ สามารถหาอ่านเพิ่มเติมได้ในฉบับที่ 150 ในหัวเรื่อง Database & Query ซึ่งผมได้กล่าวถึงหลักการและการใช้งานไว้พอสมควร สำหรับส่วนของ Report ผมยึดหลักการใช้งาน Quick Report เป็นสำคัญ เนื่องจากเป็นส่วนที่มีอยู่แล้วในรูปแบบ Native Code ภายในโปรแกรม ทำให้ค่อนข้างสะดวกเมื่อนำไปใช้งานจริง

Case Study

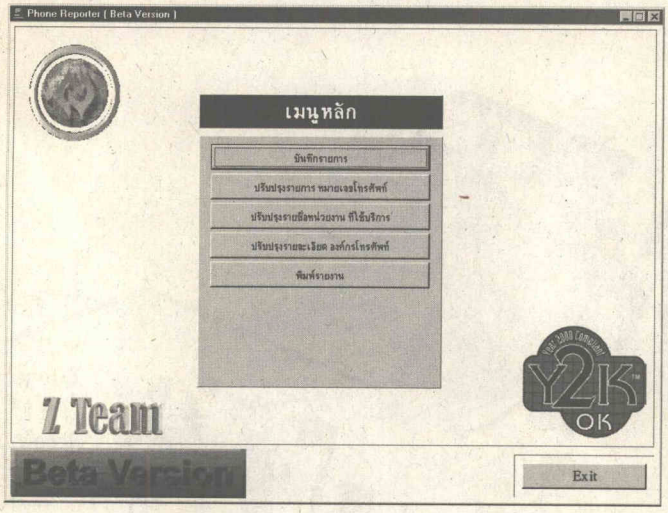
โปรแกรมสมมติที่ผมใช้เป็นตัวอย่าง และขอใช้เป็น Case Study เป็นโปรแกรมสนับสนุนการบริหารงานขององค์กร ในส่วนของการหาประสิทธิภาพการทำงานของแต่ละฝ่ายงานเปรียบเทียบกับค่าใช้จ่ายที่เกิดขึ้น

เนื่องจากหัวข้อการบริหารงานในส่วนนี้ค่อนข้างกว้าง ผมขอกำหนดไปที่งานการแจกแจงค่าใช้จ่ายการใช้โทรศัพท์ของแต่ละฝ่ายงาน เนื่องจากเป็นค่าใช้จ่ายพื้นฐานที่ทุกหน่วยงานจะต้องมี เป็นหัวเรื่องของโครงการในฉบับนี้

Phone Reporter Project

Phone Reporter Project เป็นโครงการที่มีจุดประสงค์หลักคือ แจกแจงค่าใช้จ่ายการใช้โทรศัพท์ของฝ่ายงานต่าง ๆ ในองค์กร โดยจะสรุปผลค่าใช้จ่ายการใช้โทรศัพท์ของแต่ละฝ่ายงาน

Delphi



รูปที่ 1 โปรแกรม Phone Reporter

ในทุกเดือน ออกเป็นรายงานการใช้โทรศัพท์ประจำเดือน เพื่อนำไปรวมกับผลสรุปค่าใช้จ่ายอื่น ๆ ของแต่ละฝ่าย (Cost Allocation) แล้วจึงนำไปคำนวณผลที่ต้องการในด้านต่าง ๆ เช่น ประเมินผลการทำงานของฝ่ายต่าง ๆ เป็นต้น

โปรแกรม Phone Reporter ดังรูปที่ 1 มีจุดประสงค์ของโปรแกรมคือ ทำหน้าที่บันทึกข้อมูลการใช้โทรศัพท์ของฝ่ายต่าง ๆ ในองค์กร และสรุปผลค่าใช้จ่ายโทรศัพท์ของแต่ละฝ่ายงานในแต่ละเดือนออกเป็นรายงาน โปรแกรมมีความสามารถโดยสังเขป ดังต่อไปนี้

- สนับสนุนการบันทึกข้อมูลแบ่งตามองค์กรหรือบริษัทที่ให้บริการด้านการสื่อสารต่าง ๆ เช่น องค์การโทรศัพท์แห่งประเทศไทย (TOT) เทเลคอมเอเชีย (TA) และ แอดวานซ์ อินโฟร์ เซอร์วิส (AIS) เป็นต้น
- สนับสนุนการบันทึกค่าบริการของ การสื่อสารแห่งประเทศไทย (CAT) ในกรณีโทรทางไกล หรือต่างประเทศ
- คำนวณค่าใช้จ่ายจริง ปรับตามภาษีมูลค่าเพิ่ม (VAT) ที่กำหนดไว้ในปัจจุบัน
- บันทึกเป็นค่าใช้จ่ายสำหรับฝ่ายงานต่าง ๆ โดยสามารถเพิ่มเติม หรือแก้ไขรายละเอียดของฐานข้อมูลได้
- พิมพ์รายงานสรุปการใช้โทรศัพท์ และค่าบริการขององค์การสื่อสารแห่งประเทศไทย

อนึ่ง เพื่อเข้าใจโดยง่าย ผมใช้การกำหนดหมายเลขโทรศัพท์แต่ละหมายเลข ให้กับฝ่ายงานต่าง ๆ โดยตรง โดยแต่ละฝ่ายงานจะมีหมายเลขโทรศัพท์ที่ใช้งานเป็นของฝ่ายตัวเอง ไม่จำกัดจำนวน ซึ่งในการใช้งานจริง ๆ แล้ว บริษัทขนาดกลางหรือใหญ่อาจจะใช้หมายเลขภายในที่กำหนดในตู้ชุมสายโทรศัพท์ย่อย (PABX) ของบริษัทเอง ทั้งนี้ขึ้นอยู่กับนโยบายของบริษัท หรือความสามารถของตู้ชุมสายโทรศัพท์เอง

ประโยชน์ที่ได้รับจากการศึกษาโปรแกรมนี้

เนื่องจากโปรแกรมเป็นโปรแกรมขนาดกลาง มีการเชื่อมโยงงานต่าง ๆ ผ่านฟอร์ม (Form) หลายฟอร์ม ทำให้ผู้สนใจสามารถศึกษาการใช้งานส่วนต่าง ๆ ดังต่อไปนี้

- การใช้งาน Database Query ด้วย ODBC (Open Database Connectivity) กับฐานข้อมูลชนิด MDB ของโปรแกรมจัดการฐานข้อมูล Microsoft Access
- การใช้คำสั่ง SQL สร้าง Query ด้วย Delphi และการอัพเดตข้อมูลแบบ Cached Updates
- การใช้ Query สร้างรายงานด้วย Quick Report
- การสร้าง Data module สำหรับรวบรวม Dataset และ Datasource
- การสร้าง Unit ย่อยเพื่อสะสม (Collection) โปรแกรมย่อย หรือฟังก์ชันรวมที่สามารถใช้งานร่วมกันได้
- การสร้างโปรแกรมแบบ Multi-Form

เตรียมการ

ในโครงการนี้ผมเลือกใช้ฐานข้อมูลชนิด MDB ของ Microsoft Access ที่อยู่ในชุด Microsoft Office เวอร์ชัน 97 เป็นระบบฐานข้อมูลหลัก โดยผมพิจารณาจากความยืดหยุ่นของระบบโครงสร้างไฟล์ MDB และความนิยมในการใช้งานเป็นหลัก เนื่องจากผู้สนใจบทความชุดนี้สามารถทำการสร้างแฟ้มข้อมูลชนิดนี้บนเครื่องที่ใช้ทำการศึกษาได้ทันที เพียงแต่ขอให้มีการติดตั้งชุด ODBC Driver (Open Database Connectivity) เท่านั้น โดยไม่จำเป็นต้องติดตั้งโปรแกรมจัดการฐานข้อมูลใด ๆ หรือยุ่งยากในการจัดเตรียมเครื่อง Database Server และเมื่อนำไปใช้งานจริงสามารถทำได้ทันทีโดยการทำสำเนาแฟ้มข้อมูลชนิดนี้ หรือถ้าต้องการโอนย้ายโครงสร้างข้อมูลขึ้นไปใช้งานบนระบบจัดการฐานข้อมูล Microsoft SQL Server ก็สามารถทำได้ง่าย โดยไม่ต้องแก้ไขโปรแกรมใด ๆ

เมื่อพิจารณาถึงประโยชน์ด้านอื่นที่ได้รับ จะเรียกว่าประโยชน์ทางอ้อมก็ได้ คือ การแสดงประสิทธิภาพการใช้งานด้านฐานข้อมูลในลักษณะ Independent ของ Delphi ในการเชื่อมโยงฐานข้อมูลชนิดอื่น ๆ ที่ไม่ใช่ผลิตภัณฑ์ของ Borland เอง

Delphi

สำหรับโปรแกรมและอุปกรณ์ที่ต้องจัดเตรียม ซึ่งเป็นสิ่งที่ผมคิดว่าผู้อ่านทุกท่านมีอยู่แล้ว แต่กล่าวขึ้นมาเพื่อชี้แจงให้แน่ชัด มีดังต่อไปนี้

- เครื่องคอมพิวเตอร์ พร้อมระบบปฏิบัติการวินโดวส์ 9x
- เครื่องพิมพ์ชนิดอะไรก็ได้ที่สามารถใช้บนระบบวินโดวส์ได้ (ถ้ามี)
- โปรแกรม Delphi เวอร์ชัน 3 ขึ้นไป
- ODBC Driver ซึ่งปกติจะถูกติดตั้งให้เรียบร้อยเมื่อมีการติดตั้งชุดโปรแกรม Microsoft Office 97
- โปรแกรมจัดการฐานข้อมูล Microsoft Access 97 (ถ้ามี)

เตรียมฐานข้อมูล

ในการสร้างแฟ้มข้อมูลชนิด MS-Access (MDB) จำเป็นที่ต้องใช้โปรแกรม Microsoft Access 97 (ผมขอให้ใช้เวอร์ชัน 97 เนื่องจากผมใช้เวอร์ชันเดียวกันนี้ในการสร้างแฟ้มข้อมูลตัวอย่าง) โดยถ้ามีโปรแกรมนี้ติดตั้งอยู่ในเครื่องอยู่แล้ว ก็สามารถทำการสร้างได้ทันที แต่ถ้าไม่มีอาจจะต้องขอยืมเครื่องอื่น สร้าง แล้วคัดลอกสำเนามา หรือหาซื้อโปรแกรมตัวนี้มาติดตั้งก่อนครับ

ผมออกแบบโครงสร้างฐานข้อมูลนี้ไว้เท่าที่จำเป็นก่อน 7 ตาราง (Table) โดยบันทึกลงแฟ้มข้อมูลชื่อ C:\PR\PR.MDB มีรายละเอียดดังต่อไปนี้ครับ

Department Table หรือตารางรายชื่อฝ่ายงาน เป็นตารางที่ใช้บันทึกรายชื่อฝ่ายงานทั้งหมดที่มีอยู่ในองค์กร

Department Table : ตารางรายชื่อฝ่ายงาน		
Field Name	Field Type	Meaning
RecID	AutoNumber	Auto record counter (Reserved for the future use)
Dept_No	Number (Long)	รหัสประจำฝ่ายงาน
Dept_Desc	Text (50 Char.)	ชื่อ รายละเอียดฝ่ายงาน

Monthly_Table : ตารางรายชื่อเดือนที่ใช้ในระบบ ใช้ใน Query เมื่อมีการสั่งพิมพ์รายงาน สำหรับแสดงชื่อเดือนที่เลือก

Monthly_Table : ตารางรายชื่อเดือนที่ใช้ในระบบ		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
MonthName	Text (50)	รายชื่อเดือน

Organization_Table : ตารางรายชื่อองค์กร หรือบริษัท ที่ให้บริการด้านโทรศัพท์ ระบุรายละเอียดต่าง ๆ ของแต่ละองค์กร/บริษัท เช่น ค่าธรรมเนียมรายเดือน ค่าบริการต่อนาที/ครั้ง อัตราภาษีหัก ณ ที่จ่ายสำหรับบริษัทเอกชนที่ให้บริการโทรศัพท์ (หมายเลขสำหรับโทรศัพท์ชนิด Mobile ทั้งหมด)

Organization_Table : ตารางรายชื่อองค์กร หรือบริษัท ที่ให้บริการด้านโทรศัพท์		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
OrgCode	Number (Long)	รหัสประจำองค์กร
Name	Text (50)	ชื่อเต็มองค์กร
NickName	Text (10)	อักษรย่อขององค์กร
Monthly_Fee	Number (Double)	ค่าธรรมเนียมรายเดือน
Fee_Tax	Number (Double) สำหรับบริษัทเอกชน	% ของภาษีหัก ณ ที่จ่ายสำหรับค่าธรรมเนียมรายเดือน
Call_Tax	Number (Double)	% ของภาษีหัก ณ ที่จ่ายสำหรับค่าบริการโทรศัพท์ สำหรับบริษัทเอกชน
VAT_Rate	Number (Double)	% อัตราภาษีมูลค่าเพิ่มที่ใช้งาน
Serv_Rate_Per_Min	Number (Double)	อัตราค่าบริการต่อนาที หรือต่อครั้ง

Period_Table : ตารางช่วงเวลาการเรียกเก็บค่าบริการ ในกรณีเก็บมากกว่า 1 ครั้งต่อเดือน เช่น 2 งวดต่อเดือน

Period_Table : ตารางช่วงเวลาการเรียกเก็บค่าบริการ		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
PeriodNo	Number (Long)	รหัสประจำช่วงเวลา
PeriodMeaning	Text (50 Char.)	ความหมายของช่วงเวลา

Telephone_Table : ตารางรายการหมายเลขโทรศัพท์ที่มีใช้อยู่ในองค์กรทั้งหมด รวมทั้งระบุชนิดที่ใช้ และฝ่ายงานที่เกี่ยวข้องในปัจจุบัน

Telephone_Table : ตารางรายการหมายเลขโทรศัพท์ที่มีใช้อยู่ในองค์กรทั้งหมด		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
Tel_No	Text (10 Char.)	หมายเลขโทรศัพท์ สามารถบันทึกหมายเลข Leased-Line
OrgCode	Number (Long)	รหัสประจำองค์กร
Tel_Type	Number (Long)	ระบุชนิดของโทรศัพท์ว่าใช้ทำอะไร เช่น 1=Tel, 2=FAX, 3=Modem Etc.
Dept_No	Number (Long)	รหัสประจำฝ่ายที่ใช้งานอยู่ในปัจจุบัน

Transaction_CAT_Table : ตารางบันทึกรายการ transaction ประจำของการสื่อสารแห่งประเทศไทย ซึ่งจะเป็นค่าบริการโทรทางไกล หรือค่าบริการโทรต่างประเทศ โดยบันทึกรายการตามเดือน และปีที่กำหนด

Transaction_CAT_Table : ตารางบันทึกรายการ transaction ประจำของการสื่อสารแห่งประเทศไทย		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
Month	Number (Long)	ข้อมูลประจำเดือน (1-12)
Year	Number (Long)	ข้อมูลประจำปี (25xx)
Period	Number (Long)	รหัสข้อมูลช่วงเวลา
TelNo	Text (10)	หมายเลขโทรศัพท์
Dept_No	Number (Long)	รหัสของฝ่ายที่ใช้งานในช่วงเดือนนั้น
RecDate	Date / Time	วันที่บันทึกข้อมูล
ServiceCharge	Number (Double)	ค่าบริการไม่รวม VAT
VAT	Number (Double)	ยอดเงินภาษีมูลค่าเพิ่ม
TotAmount	Number (Double)	ยอดเงินรวมทั้งสิ้น

Transaction_TOT_Table : ตารางบันทึกรายการ Transaction ขององค์กร หรือบริษัทที่ให้บริการโทรศัพท์ โดยบันทึกรายการตามเดือน และปีที่กำหนด สำหรับบริษัทเอกชนที่ให้บริการโทรศัพท์ชนิด Mobile Phone หรือโทรศัพท์มือถือ จะมีการบันทึกจำนวนเงินภาษีเงินได้ หัก ณ ที่จ่ายทั้งสำหรับค่าธรรมเนียมรายเดือน (FeeTAX) และค่าบริการรายเดือน (ServTAX)

Transaction_TOT_Table : ตารางบันทึกรายการ Transaction ขององค์กร		
RecID	AutoNumber	Auto record counter (Reserved for the future use)
Month	Number (Long)	ข้อมูลประจำเดือน (1-12)
Year	Number (Long)	ข้อมูลประจำปี (25xx)
TelNo	Text (10)	หมายเลขโทรศัพท์
Dept_No	Number (Long)	รหัสของฝ่ายที่ใช้งานในช่วงเดือนนั้น
RecDate	Date / Time	วันที่บันทึกข้อมูล
ServCharge	Number (Double)	ค่าบริการไม่รวม VAT

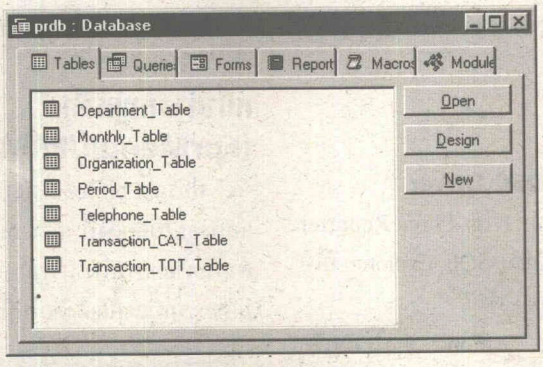
Delphi

VAT	Number (Double)	ยอดเงินภาษีมูลค่าเพิ่ม คิดตาม % ที่กำหนด
TotAmount	Number (Double)	ยอดเงินรวมทั้งสิ้น
DialCount	Number (Long)	จำนวนครั้งที่โทรออกทั้งหมด
FeeTAX	Number (Double)	จำนวนเงินภาษีหัก ณ ที่จ่ายของค่าเช่าหมายเลข (ค่าธรรมเนียมรายเดือน)
ServTAX	Number (Double)	จำนวนเงินภาษีหัก ณ ที่จ่ายของค่าบริการใช้โทรศัพท์

การสร้างฐานข้อมูล (Database)

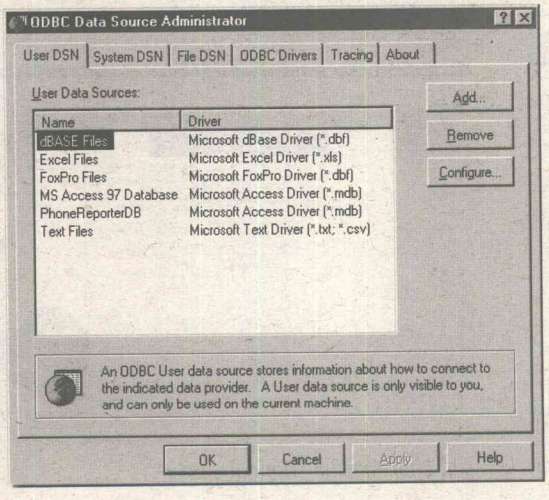
การสร้างแฟ้มข้อมูล MDB อาจจะทำได้หลายวิธี แต่สร้างด้วยโปรแกรม Microsoft Access โดยตรงจะสะดวกที่สุด สำหรับผู้ที่ไม่เคยสร้างแฟ้มข้อมูลชนิดนี้มาก่อน ผมขอกล่าวโดยสังเขปดังต่อไปนี้

- รันโปรแกรม Microsoft Access
- เลือกเมนู New Database โปรแกรมจะแสดงหน้าต่าง New เลือก Blank Database กดปุ่ม [OK]
- ตั้งชื่อแฟ้มข้อมูลเป็น C:\PRDB\PRDB.MDB กดปุ่ม [Create]
- โปรแกรมจะแสดงหน้าต่าง "PRDB : Database" เลือก Tables Page กดปุ่ม [New]
- แล้วเลือกหัวข้อ "Design View" โปรแกรมจะแสดงหน้าต่าง "Table1 : Table" ให้สร้าง Table ตามตารางทั้ง 7 ดังกล่าวข้างต้น ดังรูปที่ 2



รูปที่ 2 รายการ Tables ที่สร้างเสร็จเรียบร้อย

รูปที่ 3 การติดตั้ง PhoneReporterDB ODBC Driver



การสร้าง "PhoneReporterDB" ODBC Driver

สามารถทำได้โดยเข้าไปที่ Control Panel เลือกไอคอน 32bit ODBC (ในกรณีที่ไม่มีไอคอนนี้ต้องทำการติดตั้ง ODBC32 Driver ก่อน หรือทำการติดตั้งโปรแกรมตัวใดตัวหนึ่งของชุด Microsoft Office 97 เพื่อที่ให้โปรแกรมติดตั้งทำการติดตั้ง ODBC Driver ให้อัตโนมัติ) แล้วทำรายการดังต่อไปนี้

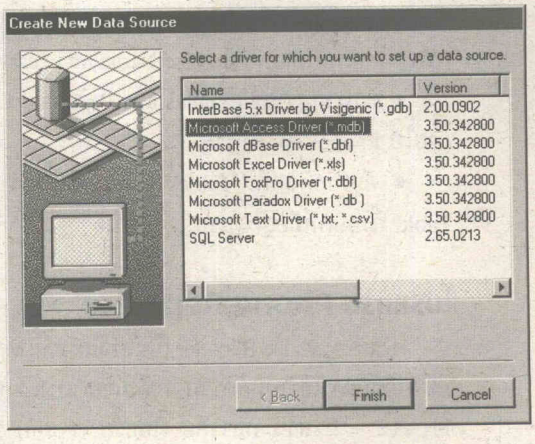
- ดับเบิลคลิกไอคอน 32bit ODBC โปรแกรมจะแสดงหน้าต่าง ODBC Data Source Administrator ดังรูปที่ 3 ภายใน Page "User DSN" จะแสดงรายชื่อ "User Data Source" ซึ่งเป็นรายชื่อ Alias ของ Datasource ที่สร้างขึ้น มีปุ่มเลือกการทำรายการ 3 รายการคือ

  – *ปุ่ม [Add]* สำหรับเพิ่มรายการ Data Source ใหม่

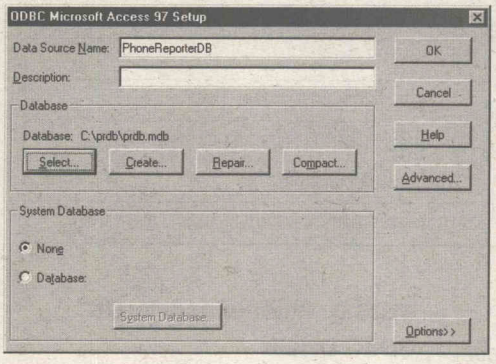
  – *ปุ่ม [Remove]* สำหรับยกเลิกรายการ Data Source ที่เลือก

  – *ปุ่ม [Configuration]* สำหรับปรับเปลี่ยน Configruation ของ Data Source ที่เลือก

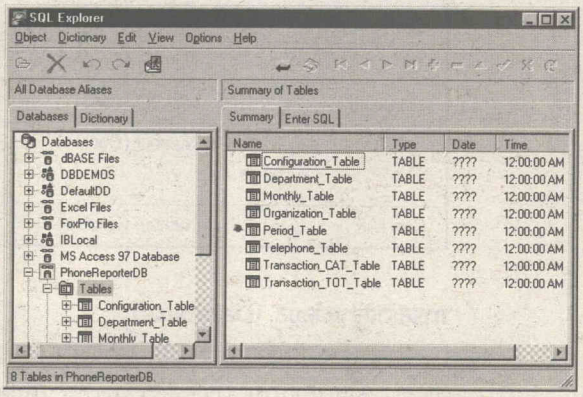
- กดปุ่ม [Add] โปรแกรมจะแสดงหน้าต่าง "Create New Data Source" ดังรูปที่ 4 ให้เลือก "Microsoft Access Driver (*.mdb)" แล้วกดปุ่ม [Finish] เพื่อติดตั้งค่า config ให้กับ Data source ตัวใหม่



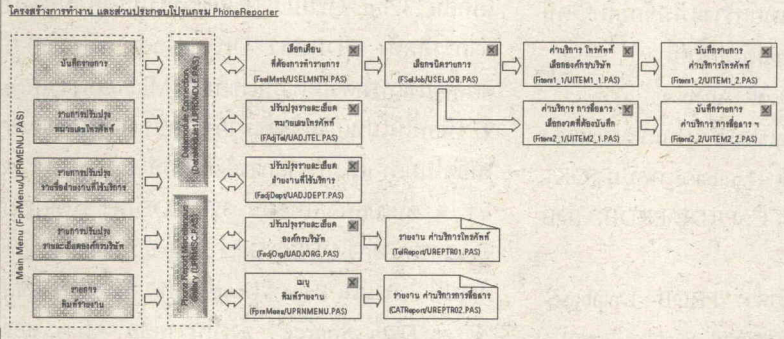
รูปที่ 4 หน้าต่าง Create New Data source



รูปที่ 5



รูปที่ 6 แสดงการติดต่อกับ PhoneReporterDB ด้วย SQL Explorer



รูปที่ 7 โครงสร้างฟังก์ชันการทำงานของโปรแกรม Phone Reporter

● โปรแกรมจะแสดงหน้าต่าง "ODBC Microsoft Access Setup" ดังรูปที่ 5 ให้ทำการตั้งชื่อ Data Source Name เป็น "PhoneReporterDB" และในกรอบ Database ให้กดปุ่ม [Select] เพื่อเลือกให้เชื่อมโยงไปยังแฟ้มข้อมูล "C:\PRDB\PRDB.MDB" กดปุ่ม [OK] เพื่อทำการบันทึก

ทดสอบการติดต่อกับ PhoneReporterDB ODBC Driver

สามารถทำการทดสอบการใช้ PhoneReporter Datasource ได้โดยใช้โปรแกรม SQL Explorer ที่ให้มาในชุดโปรแกรมของ Delphi ดังนี้

● รันโปรแกรม SQL Explorer จะปรากฏชื่อ "PhoneReporterDB" ในรายการ Database

● ดับเบิลคลิกชื่อ "PhoneReporterDB" กดปุ่ม [OK] เมื่อมีหน้าต่างถามชื่อ UserName และ Password

● จะปรากฏรายชื่อ Tables และ Procedures List ภายใต้ PhoneReporterDB

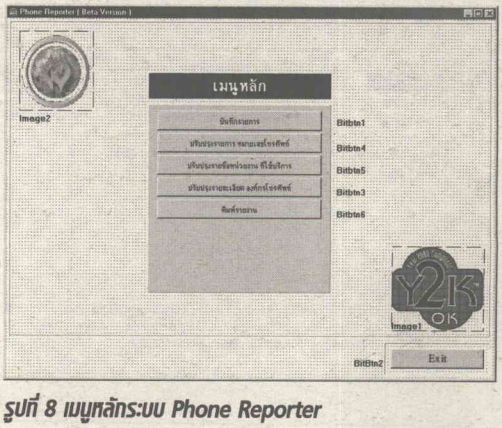
● ดับเบิลคลิกที่ "Tables List" จะแสดงรายชื่อ Table ทั้งหมดที่มีอยู่ในระบบ ดังรูปที่ 6

โปรแกรม Phone Reporter

โปรแกรม Phone Reporter เป็นโปรแกรมขนาดกลางที่มีฟังก์ชันการใช้งานหลักรวม 4 ฟังก์ชัน แบ่งแยกเป็น 11 + 2 + 2 ส่วน กล่าวคือ มีฟอร์ม (Form) ชนิดหน้าต่างรวมจำนวน 11 ฟอร์ม ฟอร์มชนิดรายงานจำนวน 2 ฟอร์ม ฟอร์มชนิด Datamodule จำนวน 1 ฟอร์ม และ Miscellanous Unit ทำหน้าที่เป็นยูนิตสะสมโปรแกรมย่อยและฟังก์ชัน ผมคงขอกล่าวถึงโดยละเอียดเฉพาะฟอร์มที่มีเนื้อหาสำคัญ และระบบการทำงานไม่ซ้ำซ้อนกับฟอร์มอื่นมากนัก เพื่อให้เนื้อหาของบทความนี้มีความกระชับไม่ซับซ้อนเกินไป สำหรับผู้ที่สนใจซอร์สโค้ดตัวเต็มของบทความนี้สามารถติดต่อใช้บริการจากซอฟต์เมล์ได้ในตอนท้ายของบทความชุดนี้ในฉบับตอนจบครับ การทำงานของระบบจะเริ่มต้นที่ Datamodule1 เป็นฟอร์มหลักและจะมีหน้าต่างเมนูหลักควบคุมการทำงานของหน้าต่างทั้งหมดอีกระดับหนึ่ง แสดงอยู่ดังรูปที่ 7

หน้าต่างเมนูหลัก (FprMenu/UPRMENU.PAS)

ฟอร์ม FprMenu ดังรูปที่ 8 มีส่วนของโปรแกรมแสดงอยู่ในลิสติงที่ 1 ทำหน้าที่เป็นหน้าต่างเมนูหลักสำหรับเรียกการทำงาน ส่วนการทำงานต่าง ๆ ของโปรแกรมมีกิจกรรมที่สำคัญดังต่อไปนี้



รูปที่ 8 เมนูหลักระบบ Phone Reporter

Delphi

ลิสติงที่ 1 UPRMENU.PAS

```
00001: // Phone Reporter
00002: // Listing 1.1 : Main Menu
00003: //
00004: unit uPrMenu;
00005: interface
00006: uses
00007:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00008:   StdCtrls, Buttons, ExtCtrls;
00009: type
00010:   TFprMenu = class(TForm)
00011:     Bevel1: TBevel;
00012:     BitBtn2: TBitBtn;
00013:     Panel1: TPanel;
00014:     BitBtn1: TBitBtn;
00015:     BitBtn3: TBitBtn;
00016:     BitBtn4: TBitBtn;
00017:     Panel2: TPanel;
00018:     BitBtn5: TBitBtn;
00019:     BitBtn6: TBitBtn;
00020:     Image1: TImage;
00021:     Image2: TImage;
00022:     Bevel2: TBevel;
00023:     Label1: TLabel;
00024:     Label2: TLabel;
00025:     Label3: TLabel;
00026:     Label4: TLabel;
00027:     Label5: TLabel;
00028:     Label6: TLabel;
00029:     Label7: TLabel;
00030:     Label8: TLabel;
00031:     procedure BitBtn2Click(Sender: TObject);
00032:     procedure FormCreate(Sender: TObject);
00033:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
00034:     procedure BitBtn1Click(Sender: TObject);
00035:     procedure BitBtn4Click(Sender: TObject);
00036:     procedure BitBtn5Click(Sender: TObject);
00037:     procedure BitBtn3Click(Sender: TObject);
00038:     procedure BitBtn6Click(Sender: TObject);
00039:   private
00040:     { Private declarations }
00041:     procedure InitVar;
00042:     procedure ReleaseVar;
00043:   public
00044:     { Public declarations }
00045:   end;
00046: var
00047:   FprMenu: TFprMenu;
00048: implementation
00049: uses uSelMnth, uprDMdle, uPrMisc, uAdjTel, uAdjDept, uAdjOrg, uPrnMenu;
00050: {$R *.DFM}
00051: //---------------------------------------------------
00052: procedure TFprMenu.InitVar;
00053: begin
00054:   curSelMonth:=0; curSelJob:=0; curSelOrg:=''; curSelOrgNum:=0;
00055:   curPeriod:=0;
00056: end;
00057: procedure TFprMenu.ReleaseVar;
00058: begin end;
00059: //---------------------------------------------------
00060: procedure TFprMenu.BitBtn2Click(Sender: TObject);
00061: begin
00062:   FprMenu.Close;
00063: end;
00064: procedure TFprMenu.FormCreate(Sender: TObject);
00065: begin
00066:   InitVar;
00067: end;
00068: procedure TFprMenu.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
```

procedure TFprMenu.BitBtn1Click(Sender: TObject); กิจกรรมสำหรับสร้างฟอร์ม FselMnth เป็นฟอร์มเริ่มต้นสำหรับทำรายการบันทึกข้อมูล ใช้เลือกเดือนที่ต้องการบันทึกข้อมูลเป็นอันดับแรก เราสามารถสร้างฟอร์มด้วยคำสั่ง

```
if ( FselMnth=nil ) then  Application.CreateForm( TFselMnth,
FselMnth );
FselMnth.Show; FprMenu.Hide;
```

(โดยปกติเมื่อมีการเพิ่มฟอร์มใหม่ลงใน Project File ฟอร์มใหม่ที่เพิ่มจะถูกกำหนดให้สร้างเองอัตโนมัติเมื่อโปรแกรมรัน การใช้คำสั่ง CreateForm เป็นการสั่งให้โปรแกรมสร้างฟอร์มใหม่ขึ้นตามความต้องการของผู้ใช้เอง การใช้งานลักษณะนี้จำเป็นต้องยกเลิกการสร้างฟอร์มอัตโนมัติ สามารถทำได้โดยยกเลิกในเมนู Project → Options → Form Page ให้ย้ายรายชื่อฟอร์มที่ต้องการจากช่อง Auto-create forms ไปไว้ใน Available forms)

procedure TFprMenu.BitBtn4Click(Sender: TObject); กิจกรรมสำหรับสร้างหน้าต่าง FadjTel สำหรับปรับปรุงรายการหมายเลขโทรศัพท์

procedure TFprMenu.BitBtn5Click(Sender: TObject); กิจกรรมสำหรับสร้างฟอร์ม FadjDept สำหรับปรับปรุงข้อมูลของฝ่ายที่ใช้หมายเลขโทรศัพท์

procedure TFprMenu.BitBtn3Click(Sender: TObject); กิจกรรมสำหรับสร้างหน้าต่าง FadjOrg สำหรับปรับปรุงข้อมูลขององค์กร/บริษัท ที่ให้บริการด้านโทรศัพท์

procedure TFprMenu.BitBtn6Click(Sender: TObject); กิจกรรมสำหรับสร้างฟอร์ม FprnMenu เป็นเมนูของส่วนระบบการพิมพ์รายงาน

ฟอร์ม FprMenu จะเรียกใช้ฟอร์ม และกิจกรรมอื่นโดยระบุชื่อยูนิต (Units) ของฟอร์มต่าง ๆ ไว้ในส่วน Uses หลัง Implementation Interface คือ

```
implementation
uses uSelMnth, uprDMdle, uPrMisc, uAdjTel, uAdjDept, uAdjOrg,
uPrnMenu;
```

Datamodule Form (DataModule1/ UPRDMDLE.PAS)

ฟอร์ม Datamodule ทำหน้าที่เก็บชุด Dataset และ Datasource ที่ใช้อยู่ในระบบ มีรายละเอียดของฟอร์ม

Delphi

ลิสติงที่ 1 (ต่อ)

```
00069:  begin
00070:    ReleaseVar;
00071:    Canclose:=true;
00072:  end;
00073:  procedure TFprMenu.BitBtn1Click(Sender: TObject);
00074:  begin
00075:    if ( FselMnth=nil ) then
00076:      Application.CreateForm( TFselMnth, FselMnth );
00077:    FselMnth.Show; FprMenu.Hide;
00078:  end;
00079:  procedure TFprMenu.BitBtn4Click(Sender: TObject);
00080:  begin
00081:    if ( FAdjTel=nil ) then
00082:      Application.CreateForm( TFAdjTel, FAdjTel );
00083:    FAdjTel.Show; FprMenu.Hide;
00084:  end;
00085:  procedure TFprMenu.BitBtn5Click(Sender: TObject);
00086:  begin
00087:    if ( FadjDept=nil ) then
00088:      Application.CreateForm( TFadjDept, FadjDept );
00089:    FadjDept.Show; FprMenu.Hide;
00090:  end;
00091:  procedure TFprMenu.BitBtn3Click(Sender: TObject);
00092:  begin
00093:    if ( FadjOrg=nil ) then
00094:      Application.CreateForm( TFadjOrg, FadjOrg );
00095:    FadjOrg.Show; FprMenu.Hide;
00096:  end;
00097:  procedure TFprMenu.BitBtn6Click(Sender: TObject);
00098:  begin
00099:    if ( FprnMenu=nil ) then
00100:      Application.CreateForm( TFprnMenu, FprnMenu );
00101:    FprnMenu.Show; FprMenu.Hide;
00102:  end;
00103:  end.
```

ลิสติงที่ 2 รายละเอียดของ Properties ต่าง ๆ สำหรับ Components ใน Datamodule1

```
object DataModule1: TDataModule1
  OldCreateOrder = True
  OnCreate = DataModule1Create
  OnDestroy = DataModule1Destroy
  Left = 198
  Top = 118
  Height = 276
  Width = 537
  object Database1: TDatabase
    AliasName = 'PhoneReporterDB'
    DatabaseName = 'PRDB'
    LoginPrompt = False
    Params.Strings = (
      'USERNAME=admin'
      'PASSWORD=')
    SessionName = 'Default'
    Left = 30
    Top = 15
  end
  object Org_DS: TDataSource
    DataSet = Org_Query
    Left = 30
    Top = 130
  end
  object Tel_DS: TDataSource
    DataSet = Tel_Query
    Left = 130
    Top = 130
  end
  object Trans_DS: TDataSource
    DataSet = Trans_Query
    Left = 345
    Top = 130
  end
  object Tel_Query: TQuery
    CachedUpdates = True
    DatabaseName = 'PhoneReporterDB'
    UpdateObject = Tel_UpdateSQL
    Left = 130
    Top = 75
  end
  object Org_Query: TQuery
    CachedUpdates = True
    DatabaseName = 'PhoneReporterDB'
    UpdateObject = Org_UpdateSQL
    Left = 30
    Top = 75
  end
  object Trans_Query: TQuery
    CachedUpdates = True
    AfterDelete = Trans_QueryAfterDelete
    DatabaseName = 'PhoneReporterDB'
    SQL.Strings = (
      'select * from transaction_tot_table')
    UpdateObject = Trans_UpdateSQL
    Left = 345
    Top = 75
  end
```

แสดงอยู่ในลิสติงที่ 2 และรายละเอียดกิจกรรมอยู่ในลิสติงที่ 3 ประกอบไปด้วย

Database1 ควบคุมการใช้ Dataset ทั้งหมดในโปรแกรม

Org_Query, Org_DS, Org_UpdateSQL เป็นชุด Dataset ที่ใช้ Query ข้อมูลองค์กร/บริษัท ที่ให้บริการโทรศัพท์ ในตาราง Organization_Table

Tel_Query, Tel_DS, Tel_UpdateSQL ชุด Dataset ที่ใช้ Query ข้อมูลโทรศัพท์ ในตาราง Telephone_Table

Dept_Query, Dept_DS, Dept_UpdateSQL ชุด Dataset ใช้ Query รายละเอียดของฝ่ายงานภายในบริษัท ในตาราง Department_Table

Trans_Query, Trans_DS, Trans_UpdateSQL ชุด Dataset ใช้ Query ทำรายการสำหรับค่าใช้บริการโทรศัพท์

CAT_Query, CAT_DS, CAT_UpdateSQL ชุด Dataset ใช้ Query ทำรายการสำหรับค่าบริการของการสื่อสารแห่งประเทศไทย

Cached Updates

ระบบการอัพเดตข้อมูลของโปรแกรม Phone Reporter ใช้วิธีการที่เรียกว่า Cached Update ดังรูปที่ 9 เป็นวิธีการเตรียมการข้อมูลที่ต้องการอัพเดตไว้ให้เรียบร้อยภายใน Internal Cache ก่อนทำการส่งไปอัพเดตข้อมูลในครั้งเดียว ซึ่งแตกต่างกับการติดต่อเพื่ออัพเดตข้อมูลในรูปแบบอื่น ที่ทุกฟิลด์จะเชื่อมโยงโดยตรงกับข้อมูลจริง ทำให้ลดขั้นตอนการติดต่อกับข้อมูล และเปิดโอกาสให้แอพพลิเคชันอื่นสามารถติดต่อกับข้อมูลได้ในขณะเดียวกัน จะมีประโยชน์อย่างมากเมื่อใช้งานกับ Remote Database Server เช่น Microsoft SQL Server

Datamodule1 มีกิจกรรมที่สำคัญดังต่อไปนี้

procedure TDataModule1.SearchDeptName (deptCode : integer); เป็นกิจกรรมสำหรับค้นหาชื่อฝ่ายงานจากฐานข้อมูลด้วย deptCode ด้วยการสร้าง SQL Statement ถ้าค้นหาไม่พบจะแสดง Messagebox แจ้งข้อความผิดพลาด

ลิสติงที่ 2 (ต่อ)

```
object Trans_UpdateSQL: TUpdateSQL
 ModifySQL.Strings = (
  'update transaction_tot_table'
 'set'
  ' month = :month,'
  ' year = :year,'
  ' telNo = :telNo,'
  ' dept_No = :dept_No,'
  ' recDate = :recDate,'
  ' ServCharge = :ServCharge,'
  ' VAT = :VAT,'
  ' TotAmount = :TotAmount,'
  ' DialCount = :DialCount,'
  ' FeeTAX = :FeeTAX,'
  ' ServTAX = :ServTAX'
  ''
  'where'
  '( month = :old_month ) and'
  '( year = :old_year ) and'
  '( telNo = :old_telNo ) ')
 InsertSQL.Strings = (
  'insert into transaction_tot_table'

   ' (month, year, telNo, Dept_No, recDate, ServCharge, VAT, TotAmo' +
   'unt, DialCount, FeeTAX, ServTAX)'
  'values'

   ' (:month, :year, :telNo, :Dept_No, :recDate, :ServCharge, :VAT,' +
   ' :TotAmount, :DialCount, :FeeTAX, :ServTAX)')
 DeleteSQL.Strings = (
  'Delete from Transaction_tot_Table'
  'where'
  ' Month = :Old_Month'
  'and'
  ' Year = :Old_Year'
  'and'
  ' TelNo = :Old_TelNo')
 Left = 345
 Top = 185
end
object Dept_Query: TQuery
 CachedUpdates = True
 DatabaseName = 'PhoneReporterDB'
 UpdateObject = Dept_UpdateSQL
 Left = 235
 Top = 75
end
object Dept_DS: TDataSource
 DataSet = Dept_Query
 Left = 235
 Top = 130
end
object CAT_Query: TQuery
 CachedUpdates = True
 DatabaseName = 'PhoneReporterDB'
 SQL.Strings = (
  'select * from transaction_cat_table')
 UpdateObject = CAT_UpdateSQL
 Left = 455
 Top = 75
end
object CAT_DS: TDataSource
 DataSet = CAT_Query
 Left = 455
 Top = 130
end
object CAT_UpdateSQL: TUpdateSQL
 ModifySQL.Strings = (
  'update transaction_cat_table'
 'set'
  ' month = :month,'
  ' year = :year,'
  ' period = :period,'
  ' telNo = :telNo,'
  ' dept_No = :dept_No,'
  ' recDate = :recDate,'
  ' ServiceCharge = :serviceCharge,'
  ' VAT = :VAT,'
  ' TotAmount = :TotAmount'
  ''
  'where'
  '( month = :old_month ) and'
  '( year = :old_year ) and'
  '( telNo = :old_telNo ) and'
  '( period =:old_period )'
  '')
 InsertSQL.Strings = (
  'insert into transaction_cat_table'

   ' (month, year, period, telNo, dept_No, recDate, ServiceCharge, ' +
   'VAT, TotAmount)'
  'values'

   ' (:month, :year, :period, :telNo, :dept_No, :recDate, :ServiceC' +
   'harge, :VAT, :TotAmount)')
 DeleteSQL.Strings = (
  'Delete from Transaction_cat_Table'
  'where'
  ' Month = :Old_Month'
  'and'
  ' Year = :Old_Year'
  'and'
  ' TelNo = :Old_TelNo'
  'and'
  ' Period = :Old_Period')
 Left = 455
 Top = 185
end
object Org_UpdateSQL: TUpdateSQL
 ModifySQL.Strings = (
  'update organization_table'
  'set'
  ' orgCode = :orgCode,'
  ' name = :name,'
  ' nickname = :nickname,'
  ' monthly_Fee = :monthly_Fee,'
  ' Fee_Tax = :Fee_Tax,'
  ' Call_Tax = :Call_Tax,'
  ' VAT_Rate = :VAT_Rate,'
  ' Serv_Rate_Per_Min = :Serv_Rate_Per_Min'
  'where'
  ' ( recid = :old_recid )')
 InsertSQL.Strings = (
  'insert into organization_table'

   ' ( orgCode, name, nickname, monthly_Fee, Fee_Tax, Call_Tax, VA' +
   'T_Rate, Serv_Rate_Per_Min )'
  'values'

   ' ( :orgCode, :name, :nickname, :monthly_Fee, :Fee_Tax, :Call_Ta' +
   'x, :VAT_Rate, :Serv_Rate_Per_Min )')
 DeleteSQL.Strings = (
  'delete from organization_table'
  'where'
  ' ( RecId = :old_recid )'
  'and'
```

Delphi

```
    ' ( name = :old_name )')
   Left = 30
   Top = 185
  end
  object Tel_UpdateSQL: TUpdateSQL
   ModifySQL.Strings = (
    'update telephone_table'
    'set'
    ' tel_No = :tel_No,'
    ' orgCode = :orgCode,'
    ' tel_type = :tel_type,'
    ' Dept_No = :Dept_No'
    'where'
    ' RecID = :old_RecID'
    ' ')
   InsertSQL.Strings = (
    'insert into Telephone_Table'
    ' ( tel_No, orgCode, tel_type )'
    'values'
    ' ( :tel_No, :orgCode, :tel_type )')
   DeleteSQL.Strings = (
    'Delete from Telephone_Table'
    'where'
    ' ( recId = :old_recid )'
    'and'
    ' ( Tel_No = :Old_Tel_No )')
   Left = 130
   Top = 185
  end
  object Dept_UpdateSQL: TUpdateSQL
   ModifySQL.Strings = (
    'update department_table'
    'set'
    ' Dept_No = :Dept_No,'
    ' Dept_Desc = :Dept_Desc'
    'where'
    ' ( recId = :old_recId )'
    ' ')
   InsertSQL.Strings = (
    'insert into department_table'
    ' ( Dept_No, Dept_Desc )'
    'values'
    ' ( :Dept_No, :Dept_Desc )')
   DeleteSQL.Strings = (
    'Delete from Department_Table'
    'where'
    ' ( recId = :old_recId )'
    'and'
    ' ( Dept_No = :old_Dept_No )')
   Left = 235
   Top = 185
  end
end
```

■

ลิสติงที่ 2 (ต่อ) ⇧

ลิสติงที่ 3 ⇨
โปรแกรม
UprDMdle.pas

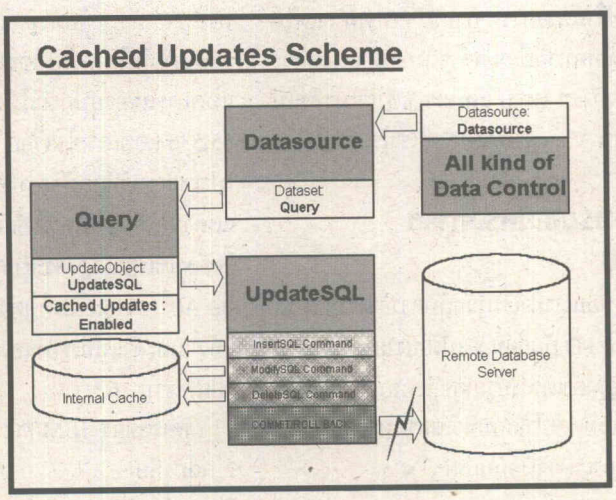
```
00001: // Phone Reporter
00002: // List 1.2 : Datamodule1
00003: //
00004: unit uprDMdle;
00005: interface
00006: uses
00007:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
00008:   Db, DBTables;
00009: type
00010:   TDataModule1 = class(TDataModule)
00011:     Database1: TDatabase;
00012:     Org_DS: TDataSource;
00013:     Tel_DS: TDataSource;
00014:     Trans_DS: TDataSource;
00015:     Tel_Query: TQuery;
00016:     Org_Query: TQuery;
00017:     Trans_Query: TQuery;
00018:     Trans_UpdateSQL: TUpdateSQL;
00019:     Dept_Query: TQuery;
00020:     Dept_DS: TDataSource;
00021:     CAT_Query: TQuery;
00022:     CAT_DS: TDataSource;
00023:     CAT_UpdateSQL: TUpdateSQL;
00024:     Org_UpdateSQL: TUpdateSQL;
00025:     Tel_UpdateSQL: TUpdateSQL;
00026:     Dept_UpdateSQL: TUpdateSQL;
00027:     procedure DataModule1Create(Sender: TObject);
00028:     procedure DataModule1Destroy(Sender: TObject);
00029:     procedure Trans_QueryAfterDelete(DataSet: TDataSet);
00030:   private
00031:     { Private declarations }
00032:   public
00033:     { Public declarations }
00034:     procedure SearchDeptName( deptCode : integer );
00035:   end;
00036: var
00037:   DataModule1: TDataModule1;
00038: implementation
00039: uses uPrMisc;
00040: {$R *.DFM}
00041: //-----------------------------------------------
00042: procedure TDataModule1.SearchDeptName( deptCode : integer );
00043: begin
00044:   with Dept_Query do
00045:   begin
00046:     close;
00047:     SQL.Clear;
00048:     SQL.add( 'select * from department_table' );
00049:     SQL.add( 'where' );
00050:     SQL.add( ' dept_no = '+IntToStr( deptCode ) );
00051:     open;
00052:   end;
00053:   Application.ProcessMessages;
00054:   if ( Dept_Query.IsEmpty ) then
00055:     ShowMessage( 'Warning : Your "selected telephone number" is not currently assign to any department' );
00056: end;
00057: //-----------------------------------------------
00058: procedure TDataModule1.DataModule1Create(Sender: TObject);
00059: begin
00060:   Database1.Connected:=true;
00061: end;
00062: procedure TDataModule1.DataModule1Destroy(Sender: TObject);
00063: begin
00064:   Database1.Connected:=false;
00065: end;
00066: procedure TDataModule1.Trans_QueryAfterDelete(DataSet: TDataSet);
00067: begin
00068:   ShowMessage( 'Current record removed.' );
00069: end;
00070: end.
```

Delphi

ลิสติงที่ 4 ไฟล์ UPRMISC.PAS ⇩

```
00001: // Phone Reporter
00002: // List 1.4 : UPRMISC.PAS
00003: //
00004: unit uPrMisc;
00005: interface
00006: const
00007:   KmonthList : array[ 1..12 ] of string[40] =
00008:     ( 'มกราคม','กุมภาพันธ์','มีนาคม','เมษายน','พฤษภาคม','มิถุนายน',
00009:       'กรกฎาคม','สิงหาคม','กันยายน','ตุลาคม','พฤศจิกายน','ธันวาคม' );
00010: const
00011:   curYear : integer = 2542;
00012: //- transaction type ----------------------------------------
00013:   NOT_SELECT    : word = $0100;
00014:   INSERT_TYPE   : word = $0101;
00015:   EDIT_TYPE     : word = $0102;
00016:   DELETE_TYPE   : word = $0103;
00017:   PRN_TEL_FORM  = $0104;
00018:   PRN_CAT_FORM  = $0105;
00019:   PRN_DEPT_FORM = $0106;
00020: var
00021:   curPeriod    : integer;
00022:   curSelMonth  : integer;
00023:   curSelJob    : integer;
00024:   curSelOrg    : shortString;
00025:   curSelOrgNum : integer;
00026: function  GetSelectMonth : string;
00027: function  GetVATValue( VATprc, VATrate : double ) : double;
00028: function  GetprcFromVATprc( VATprc, VATrate : double ) : double;
00029: implementation
00030: function GetSelectMonth : string;
00031: var ret : string;
00032: begin
00033:   ret:='';
00034:   if ( curSelMonth in [ 0..11 ] ) then ret:=KmonthList[ curSelMonth+1 ];
00035:   result:=ret;
00036: end;
00037: function GetVATValue( VATprc, VATrate : double ) : double;
00038: begin
00039:   result:=( VATprc * VATrate ) / ( 100+VATrate );
00040: end;
00041: function GetprcFromVATprc( VATprc, VATrate : double ) : double;
00042: begin
00043:   result:=VATprc - GetVATValue( VATprc, VATrate );
00044: end;
00045: end.
```

รูปที่ 9 ⇨ *การใช้ระบบ Cached Updates กับ Query*



Miscellanous Unit (UPRMISC.PAS)

UPRMISC.PAS เป็นยูนิตกลางที่รวบรวมค่าคงที่ ตัวแปร และฟังก์ชันที่มีการเรียกใช้ร่วมกันในฟอร์มต่าง ๆ ตัวโปรแกรมแสดงอยู่ในลิสติงที่ 4 มีตัวแปรและฟังก์ชันที่สำคัญดังต่อไปนี้

curPeriod : integer; // ช่วงเวลาที่เลือกสำหรับบันทึกข้อมูลของการสื่อสารแห่งประเทศไทย

curSelMonth : integer; // เดือนที่เลือกเพื่อใช้บันทึกข้อมูล

curSelJob : integer; // ชนิดรายการที่เลือกทำ

curSelOrg : shortString; // ชื่อองค์กร/บริษัทที่ให้บริการโทรศัพท์ ที่เลือก

curSelOrgNum : integer; // หมายเลขรหัสของชื่อองค์กร/บริษัทที่ให้บริการโทรศัพท์

function GetSelectMonth : string; ฟังก์ชันอ่านชื่อเดือนปัจจุบันที่เลือกจากตัวแปร curSelMonth

function GetVATValue(VATprc, VATrate : double) : double; ฟังก์ชันคำนวณหามูลค่า VAT จากค่าบริการที่รวมภาษีมูลค่าเพิ่ม VATprc โดยคำนวณตามอัตรา VATrate ด้วยสูตร

```
result:=( VATprc * VATrate ) / ( 100+VATrate );
```

function GetprcFromVATprc(VATprc, VATrate : double) : double; ฟังก์ชันคำนวณหาค่าบริการที่ไม่รวม VAT จากค่าบริการที่รวมภาษีมูลค่าเพิ่ม VATprc โดยคำนวณตาม VATrate ด้วยสูตร

```
result:=VATprc - GetVATValue( VATprc, VATrate );
```

เป็นอย่างไรบ้างครับ หวังว่าผู้อ่านทุกท่านคงเข้าใจในเนื้อหาที่ผมกล่าวถึงทั้งหมด ผมขอสรุปไว้ดังนี้คือ ส่วนแรกผมกล่าวถึง โครงสร้างและแนวความคิดของโปรแกรม Phone Reporter ตามด้วยวิธีการสร้าง Database ด้วยโปรแกรม Microsoft Access รวมทั้งการสร้าง ODBC Data source และปิดท้ายด้วยฟอร์มและยูนิตอีก 3 ส่วนที่ใช้ในระบบ ในกรณีที่ท่านผู้อ่านมีปัญหาเกี่ยวกับบทความ หรือไม่เข้าใจบทความตอนไหน สามารถติดต่อโดยตรงกับผมที่ prapona@hotmial.com สำหรับฉบับนี้คงต้องจากกันเท่านี้ก่อนครับ พบกันใหม่ฉบับหน้า สวัสดีครับ